a program interpreter for directly executing the instructions as a sequence of calls on the library so as to directly construct the complex data object from the persistent representation, without parsing the persistent representation.

10. (Three Times Amended) A system for recreating a complex data object from a persistent representation of its structure, the system comprising:

C4

a library having a predefined set of data types and methods for creating complex data objects; and

a program interpreter for interpreting the contents of the persistent representation as a sequence of directly executable instructions, and for executing those instructions as a sequence of calls on the library so as to construct the complex data object directly from the persistent representation, without parsing the persistent representation.

C5

15. (Twice Amended) A storage medium containing a persistent representation of the structure of a multicomponent data object as a sequence of instructions directly executable, wherein the directly executable instructions are calls to a set of predefined functions that are called by a program interpreter implemented in a digital computer so as to recreate the structure of the multicomponent data object, without parsing the persistent representation.

C6

19. (Three Times Amended) A storage medium containing computer-executable instructions and data for interpreting a persistent representation of the structure of a complex data object as a sequence of directly executable virtual instructions for directly constructing the complex data object as a series of calls on a library of predefined functions, without parsing the persistent representation.

Please add Claims 28-65 as follows:

C7

28. (New) A computer-implemented method for recreating a complex data object having a structure, the method comprising:

recording a sequence of calls from an authoring tool to a set of predefined functions used to create a complex data object;

translating the recorded sequence of calls into a sequence of directly executable instructions;

storing the sequence of directly executable instructions as a persistent representation of the structure of the complex data object;

obtaining the persistent representation of the structure of the complex data object and interpreting the directly executable instructions as calls to a set of predefined functions; and

calling a predefined function corresponding to each directly executable instruction in the sequence of directly executable instructions to construct the complex data object directly from the persistent representation.

C1

29. (New) A method according to claim 28, further comprising displaying the complex data object to a user.

30. (New) A method according to claim 28, wherein the complex data object is a multimedia presentation.

31. (New) A method according to claim 28, wherein at least some of the functions have arguments.

32. (New) A method according to claim 31, wherein a call to one of the functions includes a call to another function as an argument of the first function.

33. (New) A method according to claim 31, wherein a call to one of the functions includes obtaining a constant value as its argument.

34. (New) A method according to claim 28, wherein at least some of the functions return an explicit result.

35. (New) A method according to claim 28, wherein the set of predefined functions further comprises information about the readable name of each function, whether each function returns a result, whether each function has a side effect, and the number of arguments for each function.

36. (New) A method according to claim 35, wherein recording the sequence of calls further comprises obtaining the readable name of each function, whether each function returns a result, whether each function has a side effect, and the number of arguments for each function from the set of predefined functions.

37. (New) A method according to claim 36, wherein the sequence of calls are recorded as a sequence of code fragments representing each function called, the arguments to each function, and the order in which the authoring tool made the call.

C1

38. (New) A method according to claim 28, wherein the sequence of directly executable instructions are interpreted by a stack-based virtual machine.

39. (New) A computer-readable medium having computer-executable instructions for performing the method recited in claim 28.

40. (New) A computer system having a processor, a memory, and an operating environment, the computer system being operable for performing the method recited in claim 28.

41. (New) A system for recreating a complex data object from a persistent representation of its structure, the system comprising:

an authoring tool for recording a sequence of calls to a predefined set of data types and methods for creating a complex data object;

a program generator for translating the recorded sequence of calls into a sequence of directly executable instructions and storing the sequence of directly executable instructions as a persistent representation of the structure of the complex data object;

a library having a predefined set of data types and methods for creating complex data objects; and

a program interpreter for interpreting the contents of the persistent representation as a sequence of directly executable instructions, and for executing those instructions as a sequence of calls on the library so as to construct the complex data object directly from the persistent representation.

42. (New) A system according to claim 41, where the complex data object is a multimedia presentation.

43. (New) A system according to claim 41, where the program interpreter is a virtual machine located in a computer in which the complex data object is presented to the user.

C1

44. (New) A system according to claim 43, wherein the program interpreter is a stack-based virtual machine.

45. (New) A system according to claim 44, wherein the stack-based virtual machine further includes a temporary storage array.

46. (New) A system according to claim 41, wherein the predefined set of data types and methods further comprises information about the readable name of each method, whether each method returns a result, whether each method has a side effect, and the number of arguments for each method.

47. (New) A system according to claim 46, wherein recording the sequence of calls further comprises obtaining the readable name of each method, whether each method returns a result, whether each method has a side effect, and the number of arguments for each method from the predefined set of data types and methods.

48. (New) A system according to claim 47, wherein the sequence of calls are recorded as a sequence of code fragments representing each method called, the arguments to each method, and the order in which the authoring tool made the call.

49. (New) A storage medium containing a persistent representation of the structure of a multicomponent data object as a sequence of instructions directly executable, wherein the directly executable instructions are generated by recording a sequence of calls from an authoring tool to a set of predefined functions used to create the multicomponent data object and translating the recorded sequence of calls into the sequence of directly executable instructions, and wherein the directly executable instructions are calls to a set of predefined functions that are called by a program interpreter implemented in a digital computer so as to recreate the structure of the multicomponent data object.

C1

50. (New) A storage medium according to claim 49, wherein the data object is a multimedia presentation.

51. (New) A storage medium according to claim 49, wherein some of the instructions are compressed identifiers for different ones of a predefined set of methods.

52. (New) A storage medium according to claim 51, wherein others of the instructions are data in different ones of a set of predefined data types.

53. (New) A computer-implemented method for recreating a complex data object having a structure, the method comprising:

recording a sequence of calls from an authoring tool to a set of predefined functions used to create a complex data object;

translating the recorded sequence of calls into a sequence of directly executable instructions; and storing the sequence of directly executable instructions as a persistent representation of the structure of the complex data object.

54. (New) A method according to claim 53, wherein the complex data object is a multimedia presentation.

55. (New) A method according to claim 53, wherein the set of predefined functions further comprises information about the readable name of each function, whether each function returns a result, whether each function has a side effect, and the number of arguments for each function.

56. (New) A method according to claim 55, wherein recording the sequence of calls further comprises obtaining the readable name of each function, whether each function returns a result, whether each function has a side effect, and the number of arguments for each function from the set of predefined functions.

C1

57. (New) A method according to claim 56, wherein the sequence of calls are recorded as a sequence of code fragments representing each function called, the arguments to each function, and the order in which the authoring tool made the call.

58. (New) A method according to claim 53, further comprising using a stack-based virtual machine for interpreting the directly executable instructions as calls to a set of predefined functions.

59. (New) A method according to claim 53, further comprising calling a predefined function corresponding to each directly executable instruction in the sequence of directly executable instructions to construct the complex data object directly from the persistent representation.

60. (New) A computer-readable medium having computer-executable instructions for performing the method recited in claim 53.

61. (New) A system for recreating a complex data object from a persistent representation of its structure, the system comprising:

an authoring tool for recording a sequence of calls to a predefined set of data types and methods for creating a complex data object; and

a program generator for translating the recorded sequence of calls into a sequence of directly executable instructions and storing the sequence of directly executable instructions as a persistent representation of the structure of the complex data object.

62. (New) A system according to claim 61, where the complex data object is a multimedia presentation.

C1

63. (New) A system according to claim 61, wherein the predefined set of data types and methods further comprises information about the readable name of each method, whether each method returns a result, whether each method has a side effect, and the number of arguments for each method.

64. (New) A system according to claim 63, wherein recording the sequence of calls further comprises obtaining the readable name of each method, whether each method returns a result, whether each method has a side effect, and the number of arguments for each method from the predefined set of data types and methods.

65. (New) A system according to claim 64, wherein the sequence of calls are recorded as a sequence of code fragments representing each method called, the arguments to each method, and the order in which the authoring tool made the call.